locations not specified are used but have no clear one function definition

PET PAGE ZERO MEMORY MAP

FROM	TO	DESCRIPTION
000	--	$4C constant (6502 JMP instruction).
001	002	USR function address lo, hi
Terminal I/O maintenance		
003	--	Active I/O channel #.
004	--	Nulls to print for CRLF (unused).
005	--	Column BASIC is printing next.
006	--	Terminal width (unused).
007	--	Limit for scanning source columns (unused).
008	--	Line number storage before buffer.
009	--	$2C constant (special comma for INPUT process).
010	089	BASIC INPUT buffer (80 bytes).
090	--	General counter for BASIC
091	--	$00 used as delimeter.
092	--	General counter for BASIC.
Evaluation of variables		
093	--	Flag to remember dimensioned variables.
094	--	Flag for variable type: 0=numeric; 1=string.
095	--	Flag for integer tape.
096	--	Flag to crunch reserved words (protects "& remark).
097	--	Flag which allows subscripts in syntax.
098	--	Flags INPUT or READ.
099	--	Flag sign of TAN.
100	--	Flag to suppress OUTPUT (+ normal;-suppressed).
101	--	Index to next available descriptor.
102	103	Pointer to last string temporary lo; hi.
104	111	Table of double byte descriptors which point to variables.
112	113	Indirect index #1 lo; hi.
114	115	Indirect index #2 lo; hi.
116	121	Pseudo register for function operands.
Data storage maintenance		
122	123	Pointer to start of BASIC text area lo; hi byte.
124	125	Pointer to start of variables lo; hi byte.
126	127	Pointer to array table lo; hi byte.
128	129	Pointer to end of variables lo; hi byte.
130	131	Pointer to start of strings lo; hi byte.
132	133	Pointer to top of string space lo; hi byte.
134	135	Highest RAM adr lo; hi byte.
136	137	Current line being executed. A zero in 136 means statement executed in a direct command.
138	139	Line # for continue command lo; hi.
140	141	Pointer to next STMNT to execute lo; hi.
142	143	Data line # for errors lo; hi.
144	145	Data statement pointer lo; hi.
Expression evaluation		

FROM	TO	DESCRIPTION
146	147	Source of INPUT lo; hi.
148	149	Current variable name.
15Ø	151	Pointer to variable in memory lo; hi.
152	153	Pointer to variable referred to in current FOR-NEXT
154	155	Pointer to current operator in table lo; hi.
156	--	Special mask for current operator.
157	158	Pointer for function definition lo; hi.
159	16Ø	Pointer to a string descriptor lo; hi.
161	--	Length of a string of above string.
162	--	Constant used by garbage collect routine.
163	--	$4C constant (6502 JMP inst).
164	165	Vector for function dispatch lo; hi.
166	171	Floating accumulator #3.
172	173	Block transfer pointer #1 lo; hi.
174	175	Block transfer pointer #2 lo; hi.
176	181	Floating accumulator #1. (USR function evaluated here).
182	--	Duplicate copy of sign of mantissa of FAC #1.
183	--	Counter for # of bits to shift to normalize FAC #1.
184	189	Floating accumulator #2.
19Ø	--	Overflow byte for floating argument.
191	--	Duplicate copy of sign of mantissa.
192	193	Pointer to ASCII rep of FAC in conversion routine lo; hi.
RAM subroutines		
194	199	CHRGOT RAM code. Gets next character from BASIC text.
2ØØ	--	CHARGOT RAM code regets current characters.
2Ø1	2Ø2	Pointer to source text lo; hi.
203	223	Next random number in storage.
OS page zero storage		
224	225	Pointer to start of line of cursor loc lo; hi.
226	--	Column position of cursot.
227	228	General purpose start address indirect lo; hi.
229	233	General purpose and address direct lo; hi.
234	237	Flag for quote mode on/off.
238	--	Current file name length.
239	--	Current logical file number.
24Ø	--	Current primary address.
241	242	Current secondary address.
243	244	Pointer to start of current tape buffer lo; hi.
245	--	Current screen line #.
246	--	Data temporary for I/O.
247	248	Pointer to start loc for O.S. lo; hi.
249	25Ø	Pointer to current file name lo; hi.

PET PAGE TWO MEMORY MAP

FROM	TO	DESCRIPTION
512	514	24-hour clock in 1/60 sec.
517	518	Correction factor for clock LSB; MSB
519	520	Interrupt driver flag for cassette #1, switches; #2 switches.
523	--	Flag ≠ means verify not load into memory.
524	--	I/O status byte.
525	--	Index into keystroke buffer.
526	--	Flag to indicate reverse-field on.
527	536	Interrupt driven key stroke buffer.
537	538	IRQ RAM VECTOR lo; hi.
539	540	BRK instruction RAM VECTOR lo; hi.
549	--	Count down to flip cursor.
551		Flag for cursor on/off.
553	577	Table of LSB of start addresses of video display lines (25).
578	587	Table of logical addresses.
588	597	Table of primary addresses.
598	609	Table of secondary addresses.
610	--	Index into LA, FA, SA tables.
611	--	Default input device #.
612	--	Default output device #.
613	--	Computation of parity on cassette write.
621	--	Count of redundant tape blocks.
624	--	Count down synchronization or cassette write.
625	626	Index next character in/out tape buffer #1; #2.
627	--	Countdown synchronization on cassette read.
628	--	Flag to indicate bit/byte tape error.
629	--	Flag to indicate tape routine reading shorts.
630	631	Index to addresses to correct on tape read pass 1; pass 2.
632	--	Flag for cassette read--tells current function--countdown, read, etc.
633	--	Count of seconds of shorts to write before data.
634	825	Buffer for cassette #1 (192 bytes).
826	1017	buffer for cassette #2 (192 byres).
1018	1023	Unused.

FROM	TO	DESCRIPTION
251	254	Unused.
255	--	Overflow byte that BASIC uses when doing FAC to ACII conversions.

Page 1

62 bytes on bottom are used for error correction in tape reads., Also, buffer for ASCII when BASIC is expanding the FAC into a printable number. The rest of page 1 is used for storage of BASIC GOSUB and FOR NEXT context and hardware stack for the machine.

Block Ø By 256 Byte Pages

PAGE	TYPE	START ADDRESS	FUNCTION
**Ø	RAM	ØØØØ	BASIC OS working storage
**1	RAM	Ø1ØØ	Stack
**2	RAM	Ø2ØØ	O S working storage
**3	RAM	Ø3ØØ	Cassette buffers.
4-15	RAM	Ø4ØØ	BASIC text area

** see expanded description by page

Block 14 By 2K Segments

PAGE	TYPE	START ADDRESS	FUNCTION
Ø	ROM	$EØØØ	Screed editor
1	I/O	$E8ØØ	PET I/O

I/O Device Base Addresses

PAGE	TYPE	START ADDRESS	FUNCTION
Ø	PIA	$E81Ø	Keyboard
1	PIA	$E82Ø	IEEE-488
2	VIA	$E84Ø	USR port cassette

PET MEMORY ALLOCATION BY 4K BLOCKS

	BLOCK #	TYPE	START ADDRESS	FUNCTION
*	Ø	RAM	$ØØØØ	Working, text, variable storage.
	1	RAM	$1ØØØ	Test variable storage (8K only)
	2	---	$2ØØØ	Expansion RAM
	3	---	$3ØØØ	Expansion RAM
	4	---	$4ØØØ	Expansion RAM
	5	---	$5ØØØ	Expansion RAM
	6	---	$6ØØØ	Expansion RAM
	7	---	$7ØØØ	Expansion RAM
	8	RAM	$8ØØØ	Screen memory (1K)
	9	---	$9ØØØ	Expansion ROM
	1Ø	---	$AØØØ	Expansion ROM
	11	---	$BØØØ	Expansion ROM
	12	ROM	$CØØØ	BASIC (principally statement interpreter).
	13	ROM	$DØØØ	BASIC (principally math package).
*	14	ROM I/O	$EØØØ $EØØØ	Screed editor. All internal PET I/O.
	15	ROM	$FØØØ	OS diagnostics

*see expanded description